JSWITCH/JSAT: REAL-TIME AND OFFLINE WORLD WIDE WEB INTERFACE

Abigail H. Maury
Computer Sciences Corporation
7700 Hubble Drive
Lanham-Seabrook, Maryland 20706 USA
e-mail: amaury@csc.com

Anna Critchfield, CSC, *e-mail*: acritchf@csc.com
Jim Langston, CSC, *e-mail*: jlangsto@csc.com

Abstract

Jswitch, a Java-based spacecraft Web interface to telemetry and command handling, is a prototype, platform-independent user interface to a spacecraft command and control system that uses Java technology, readily available security software, standard World Wide Web (WWW) protocols, and commercial off-the-shelf (COTS) products. Jsat, a Java-based science analysis and trending tool, is a major element in Jswitch. Both Jswitch and Jsat were developed for NASA's Goddard Space Flight Center (GSFC). Jsat provides Web interface, user access to science instrument data by sending the processed science satellite trend data (user-specified graphics, tables, and reports) as Java applets via the Web to the user upon request. Jsat reduces ground science instrument data processing costs and decreases processing time by eliminating the need for special operations personnel assigned to process science instrument data requests, run trend analysis software, and collect and distribute results. Jswitch reduces operational costs by providing access to spacecraft telemetry and command from any standard Web browser.

Key words: Spacecraft Web Interface, Telemetry and Command Handling Web Interface, Science Data Trend Analysis Web Interface.

Introduction

NASA's Goddard Space Flight Center (GSFC) is funding a research effort in the use of the latest technology to reduce satellite ground system operations costs. This effort has produced the Jswitch/Jsat system, which enables a spacecraft/flight operations engineer to monitor and control a spacecraft, or enables a scientist to select and analyze the spacecraft science instrument data, over the open Internet using the standard World Wide Web (WWW) tools, browsers, and commercial-off-the-shelf (COTS) products, in conjunction with the Java programming technology. Encryption, certification, firewall, and intrusion detection technology provide security.

Jswitch implements a generic mechanism by downloading applets to access mission control center systems over the Internet. The Jswitch system interfaces through a generic bridge to COTS control center systems. For example, we process telemetry pages through an interface to the following systems: EPOCH 2000, Transportable Payload Operations Control Center (TPOCC), Generic Spacecraft Analyst Assistant (GenSAA), and the Advanced System for Integration and Spacecraft Test (ASIST). Jsat, a major element in Jswitch, interfaces with the Archive Browser Extractor (ABE), a COTS non-real-time spacecraft telemetry data analysis package that provides the statistical subset data. Integral Systems, Inc. (ISI) developed EPOCH 2000 and ABE and provided both products to NASA/GSFC for prototyping the Jswitch/Jsat system. Currently, GSFC is integrating components of the Jswitch system into the OMNI (Operating Mission as Nodes on the Internet) system for the Solar Eclipse Mission to the Black Sea in August 1999. OMNI will transmit weather, Global Positioning System (GPS) data, and eclipse images to museums, schools, and the public over the Internet.

The Jswitch/Jsat system demonstrates how a combination of current technologies can be used in mission support systems today. The Jswitch system reduces operations costs through operational scenarios that enable remote diagnosis of problems detected during lights-out operations periods. The Jsat system removes the necessity for operators to provide data to scientists by giving scientists direct access through the Web. The system allows distributed operations teams to use the inexpensive open Internet for wide area access.

The Jswitch/Jsat system provides flexibility for use across many mission profiles by using COTS products to reduce development costs. The net result is the first opportunity to standardize user interfaces to missions, provide flexible networking, reduce operations, and significantly decrease life-cycle costs.

Jswitch/Jsat Advantages

The Java-based Jswitch software interfaces with EPOCH 2000, TPOCC, GenSAA, ASIST, or other ground systems through an application programming interface (API) using C code or Java. Using a distributed architecture, the control center software can be replaced relatively easily while the Jswitch user interface stays the same. The Jswitch/Jsat prototype gives the user the ability to view events and telemetry data, send commands, and analyze selected statistical science data—all through the use of a Web browser as the user interface. From our experience, this approach offers a number of advantages:

1. Distributed workload. Once the Java applets are downloaded to the client machine from the connected host, the applets run locally on the client's machine.
2. Portability. The Java applets will run on any machine that has a Web browser that supports Java—without any required source code changes for a specific platform (one version for all).
3. Easy user access. There is no complicated setup involved or additional software to buy for clients—the only requirement is a Web browser that supports Java and an Internet connection.
4. Easier software maintenance. Clients always connect through the central Web server, which automatically downloads the applets, making it easier to install updates to the software.
5. Direct access to the data by the scientific community. Using Jswitch/Jsat security mechanisms, user privileges can be set up to allow direct access to data for analysis without jeopardizing command and control security.
6. Availability of legacy telemetry displays via x-server Internet plug-in. The original display definitions are unchanged.
7. Generic control center interface. All mission interfaces have the same look and feel from a single Web page.

Jswitch/Jsat Web Security

Security over the open Internet is recognized as the major risk in satellite operations, particularly in commanding the satellite. Jswitch employs a number of readily available security tools and procedures to provide a secure interface. The heart of the secure interface consists of the Stronghold Web server, which is a Secure Socket Layer (SSL) implementation of the Apache Web server. Netscape's SSL, a de facto standard now, encrypts all transmissions, making it difficult for the data sent between the user's Web browser and the Web site to be monitored in transit. The encryption strength used depends on the Web browser version installed by the user. The Web server's function is to serve Java applets to the user. These Java applets provide for a graphical dynamic interface to the ground system. The applets connect to data servers to send/receive requests and data to interface programs that talk to the actual ground system. The Java applets, once downloaded on the client's machine, will run locally—only connecting back to the host for data and/or command information. The Stronghold product is easy to use, which is important because errors in complicated setups could compromise the use of the product.

This approach, as installed in the Jswitch EPOCH 2000 prototype, is to pass the data packets through the Web server and use the Web server's encryption capability (see Figure 1). The key to using server/browser encryption is to provide the data to the applet in a standard output (for telemetry) or standard input (for commands) stream via a cgi script. For data moving from the server to the browser, the applet asks to get a stream object (i.e., creates a connection to the standard output of a cgi script) using a statement such as:

```
stream = URL “https://
jswitch.nascom.nasa.gov/NPH_event.cgi ? sim1”
```

The standard output stream is established and comes from the NPH_event.cgi script. The Web server encrypts because the https:// directive is used. The parameter sim1 is used in our configuration to pass to EPOCH 2000 to ask for the data source sim1. The NPH_event.cgi script contains a call to event.c that in turn connects to EPOCH 2000. After the applet establishes the stream, the applet iterates references to stream.getline() to retrieve each packet of data that is encrypted by the server and decrypted by the browser.

The SSL Web server also provides support for digital identification, used to help assure that clients are connecting to the actual Web site and not a “bogus” site. The Jswitch Web site uses a digital certificate from Verisign. We employ a form of “firewall” software on our Jswitch machine called TCP Wrapper. This freely

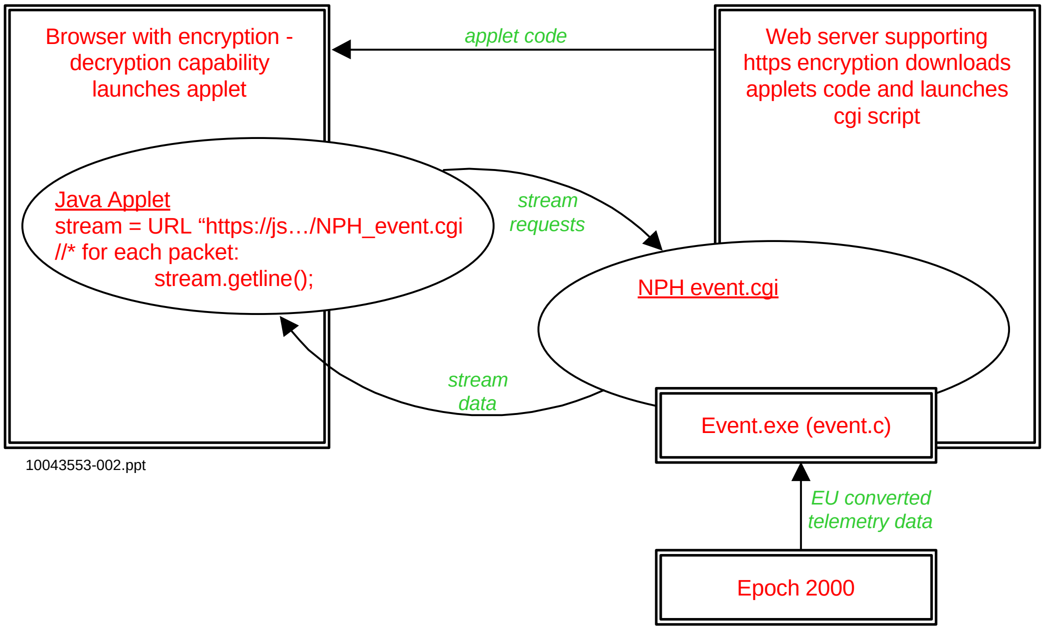

Browser with encryption - decryption capability launches applet
applet code
Web server supporting https encryption downloads applets code and launches cgi script
Java Applet
stream = URL "https://js…/NPH_event.cgi
//* for each packet:
stream.getline();
stream requests
NPH event.cgi
stream data
Event.exe (event.c)
EU converted telemetry data
Epoch 2000
10043553-002.ppt

Figure 1. Encrypted_Streaming_Objects Architecture

available tool acts as a "wrapper" for common Transmission Control Protocol (TCP) services. It provides for more detailed logging and allows for a host access and deny list for TCP services (ftp, telnet, etc.), all of which are traditionally used for remote access of a machine.

The Jswitch machine employs a form of intrusion detection software called SWATCH, freely available on the Internet. The SWATCH tool monitors the system logs and watches for specified patterns that indicate some potential foul play (such as login attempts from unauthorized users). Free probing software, called SATAN, used to check the host machine for weaknesses, was installed on another host to scan the targeted Jswitch machine for security weaknesses. Additionally, alteration of some system configuration files and services on our UNIX machine has led to a considerable improvement in system security. The primary modifications involved disabling services that are enabled in the Internet configuration file (/etc/inetd.conf on most systems).

Finally, the Jswitch machine configuration itself offers a simple, easily maintainable security solution. All the software and hardware reside on one standard UNIX machine, connected to the open Internet, but the host machine could easily be set up to have no connection to any other internal network. The only needed connection is for the ground system software to access a socket, which has the connection to data or the actual front end of a ground system. In this way, the data or front-end machines can still be kept behind whatever firewall in which they currently reside.

Operational Scenarios

By providing a uniform user interface and security mechanism, the Jswitch/Jsat architecture supports all typical operational scenarios scripted for users with different levels of operational privileges. Jswitch/Jsat naturally integrates into GSFC's use of multiple redundant strings to support satellite contacts. The Jswitch concept supports important operational scenarios that are needed to move closer to reduced operations staffs and lights-out operations during some shifts and to eliminate dedicated operations areas for small missions, enabling operators to work from their desktop computers. The following paragraphs describe some typical operational scenarios.

Anomaly Resolution Scenario: The ground system, during non-staffed off-shift hours, receives data and performs data analysis. If the ground system detects an anomaly, the system phones or pages operations personnel and/or engineers. The operations personnel from their home or office connect to the ground system over the Internet to diagnose problems by examining telemetry from recent or current passes. The operations personnel prepare and issue a real-time spacecraft command or directive that will be sent to the spacecraft during the next contact to resolve the anomaly, maintaining spacecraft health and safety.

Trend Analysis Operational Scenario: The intended user is the scientific community or engineers/analysts. User privileges are limited to playback and analysis of history data on the offline string. The user logs into the Jswitch system; the system recognizes limited privileges and connects to the offline EPOCH/ABE string. The user requests generation of statistical analysis of specified mnemonics. The Jsat component passes the request to ABE, and ABE returns the results to the user over the Internet in textual or graphical dynamic displays or formatted reports. This example shows how easy trend analysis becomes when using Jsat, compared to a legacy control center that involves time-consuming subset generation by the operations team and then distribution of correct subsets to multiple users.

Distributed Operations Scenario: Ground system servers are located in a closet at a convenient location for configuration control and hardware maintenance. Backup system(s) could be located elsewhere for geographical security. Operations personnel and spacecraft engineers perform operations duties from their desktop PCs in their offices during normal work situations and hours. Operations can be performed from remote sites during non-typical situations (e.g., conferences or meetings at other centers). Scientists and other researchers can perform science planning, receive science observations, and analyze this information from their own offices.

Jswitch/Jsat Web Interface Architecture

Figure 2 illustrates the Jswitch/Jsat architecture. ABE was the most recent addition to Jswitch to create a Jsat component that conceptualizes direct user science instrument data processing and analysis. This Web interface works as follows:

1. A Web server runs on the application host machine for remote clients to access.
2. Data server applications running on the application host machine receive socket requests from the applets and send the response back, via a socket, to the connecting Java applet. Another version uses the server encryption to return data packets via a cgi script/standard output interface.
3. The user interface is a Web browser (like Netscape Navigator) in which the user enters the Uniform Resource Locator (URL) for the Web site. This will produce the Web page and download the Java applets from the server to the Web browser on any platform. (Note: Applets are not reloaded when they are already in the browser cache.)
4. The Java applets pop up in their own frames, allowing users to enter commands, view real-time event data, monitor telemetry data, and talk to each other.

The EPOCH 2000 system and the ABE data extractor are integrated with other COTS products as a mission operations control server running on the UNIX workstation. COTS command and control systems, for example, EPOCH 2000, provide real-time playback or simulated telemetry and events in the spacecraft downlink to the Jswitch applets. The commands for uplink are received by EPOCH 2000, which performs the criticality and constraints checking and subsequently uplinks the commands to the spacecraft. The Jswitch Java code interfaces with the EPOCH 2000 system API (C interface routines) for telemetry, command, and events. The Jsat system will use a standard text editor to generate a profile file that contains all of the information that the ABE needs to generate an ABE trend table and store the data in a file for retrieval by the Jsat system. A Jsat applet retrieves the data for display on any platform with a Java-enabled browser. The Jswitch/Jsat system has a current set of tools for encryption, certification, and authentication to protect client-side security, server-side security, and document confidentiality.

COTS Products

Jswitch Software: The current COTS products in the Jswitch prototype suite are as follows: Stronghold Web server by C2net (http://www.c2.net), Digital id by Verisign (http://www.verisign.com), Java Development Kit (free, http://www.javasoft.com/products), Jswitch Java source code (free, http://moca.nascom.nasa.gov), SSLava Toolkit by Phaos (http://www.phaos.com), ground system interface software (free), and Jchart library by KLG group (http://www.klg.com.jclass/overview.html).

Software and configurations to make the client machine more secure include the following: TCP Wrapper, SWATCH, SATAN, SSH, modifications to /etc/inetd.conf services, and Secure ID by Security Dynamics (http://securitydynamics.com). These are downloadable from various sources on the Internet; one central source is ftp://coast.cs.purdue.edu/pub/tools/unix/.

The COTS control system is EPOCH 2000 by ISI. Legacy, custom control systems are TPOCC, GenSAA, and ASIST.

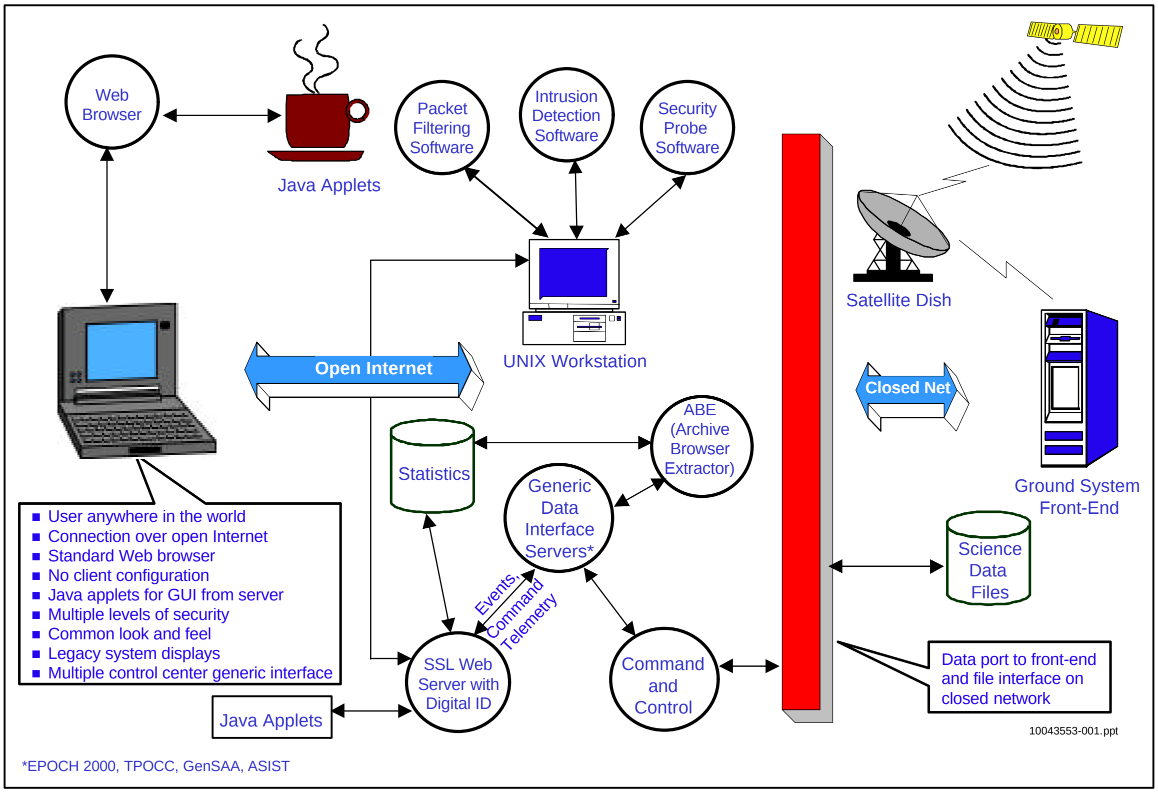

Web Browser
Java Applets
Packet Filtering Software
Intrusion Detection Software
Security Probe Software
UNIX Workstation
Satellite Dish
Open Internet
Closed Net
Ground System Front-End
Statistics
ABE (Archive Browser Extractor)
Generic Data Interface Servers*
Science Data Files
Events, Command Telemetry
SSL Web Server with Digital ID
Command and Control
Java Applets
User anywhere in the world
Connection over open Internet
Standard Web browser
No client configuration
Java applets for GUI from server
Multiple levels of security
Common look and feel
Legacy system displays
Multiple control center generic interface
Data port to front-end and file interface on closed network
10043553-001.ppt
*EPOCH 2000, TPOCC, GenSAA, ASIST

Figure 2. Java-Based Spacecraft Web Interface to Telemetry & Command Handling

Jswitch Hardware: Clients can theoretically use any Java-aware Web browser. Due to differences in some vendors' Web browsers, we find that Netscape Navigator 4.6 works the best. This is free and can be downloaded from http://www.netscape.com. The host machine, needed to serve the Java applets, run the Web server, interface to the ground system, and run the interface servers, has been Sun's UltraSparc 2, running the Solaris 2.6 operating system. The host machine does not need a Sun box as long as it is running some form of UNIX. The only caveat is that the machine should have an adequate amount of memory and processing speed to handle all the connecting clients.

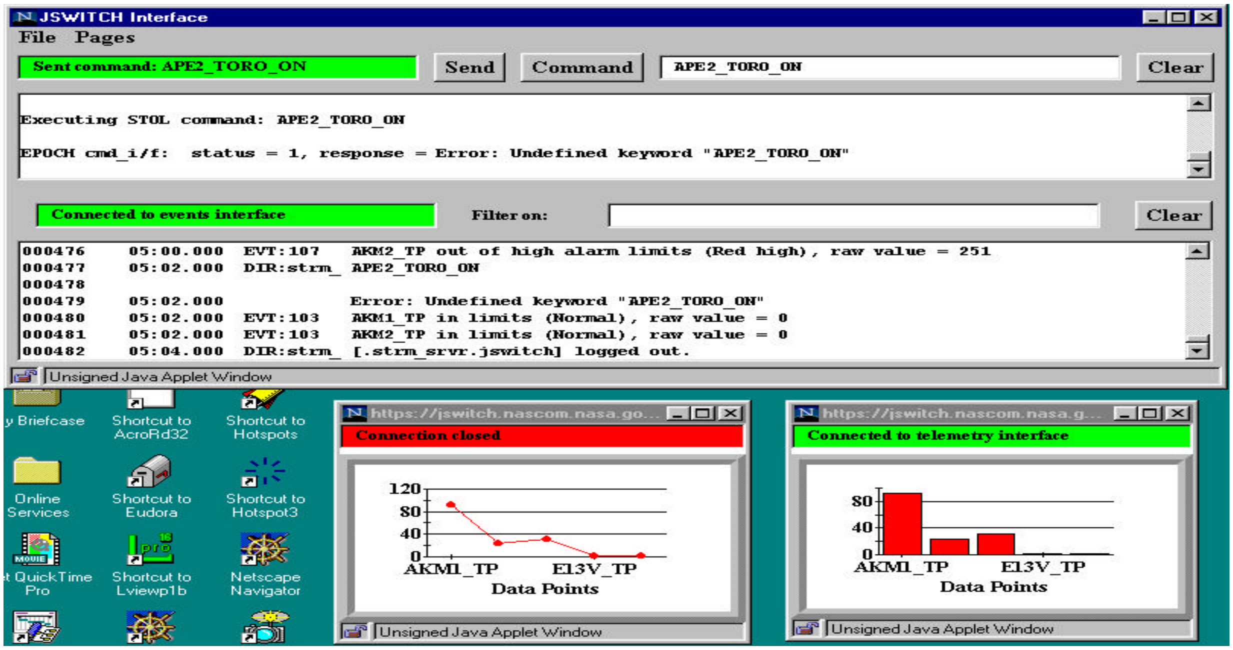
Jswitch/Jsat Implementation

A Web server runs on a host machine to provide access to clients. Users, from the server's Web page, opening their Web browsers and connecting to the host site, can select which Jswitch/Jsat functions to invoke (command, events, telemetry, or trend analysis). Selecting a function will invoke a Java applet, which will be downloaded to the client machine and then run locally. The Java applet will pop up a window, providing a user interface for the specified function. The applet will make a socket connection back to the host machine to send and receive data from a data server that is running on the host machine.

Jswitch

Figure 3, a screen snapshot, shows the Jswitch command, telemetry, and event applet running on a PC with the selected telemetry plot applets displayed. The command applet provides the user with a text entry field for entering commands to the ground system, as well as a menu selection of common commands that can be inserted by clicking. Once a command has been entered, clicking on the send button will send the command to the host machine, where the data server will send the command to the ground system, get the result back, and then return it to the command applet. The command applet will display the original command, along with the result, in a scrolling text area. The command action is entered into the scrolling event display.

The event applet provides the user with an event list from the ground system. Once activated, it will connect to a data server, which fetches events from the ground system. These events are displayed in the event applet in a scrolling text area. Event messages are color-coded



Sent command: APE2_TORO_ON

APE2_TORO_ON

```
Executing STOL command: APE2_TORO_ON

EPOCH cmd_i/f:  status = 1, response = Error: Undefined keyword "APE2_TORO_ON"
```

Connected to events interface

```
000476    05:00.000  EVT:107   AKM2_TP out of high alarm limits (Red high), raw value = 251
000477    05:02.000  DIR:strm_ APE2_TORO_ON
000478
000479    05:02.000            Error: Undefined keyword "APE2_TORO_ON"
000480    05:02.000  EVT:103   AKM1_TP in limits (Normal), raw value = 0
000481    05:02.000  EVT:103   AKM2_TP in limits (Normal), raw value = 0
000482    05:04.000  DIR:strm_ [.strm_srvr.jswitch] logged out.
```

Connection closed

Connected to telemetry interface

Figure 3. Jswitch: Telemetry and Command Applet

(red, out of limits; yellow, starting to exceed limits;green, normal limits; and white, spacecraft events). The user can filter events.

The telemetry function provides the user with a list of directories and pages to select. These pages are American Standard Code for Information Interchange (ASCII) text files residing on the host machine that describe the appearance of a telemetry page and a list of data points. Currently, the displays consist of basic text displays and graphical displays. Additional graphical displays are available by using COTS products, which provide Java graphical libraries. Once the user selects a desired page, this information is sent to a data server, where data updates are sent to the telemetry applet, which displays the telemetry information in a window. The users may define their mnemonics for graphical display.

Jswtich/X-Protocol Display Enhancement: Jswitch has added an option to make viewing of the Gamma Ray Observatory (GRO) legacy telemetry pages via an x-server Internet plug-in running on a desktop PC. Historically, spacecraft mission telemetry pages are customized for each spacecraft, and conversion to a different format is costly, involving retraining operations personnel. By providing legacy mission displays with no changes, operations can continue with no impact and without extra costs, and the chances of operational errors associated with changing existing displays are reduced.

Jswitch/Generic Bridge Enhancement: Jswitch is developing a common interface to connect to multiple control centers. This approach has been demonstrated to work in the connection to EPOCH 2000, TPOCC, and the GenSAA toolkit. We are currently prototyping connections to other mission systems such as GSFC's ASIST. The common Web interface will present multiple control center interfaces associated with privileges to access the data for operations personnel and schools.

Jsat

Jsat interfaces with ABE, the ISI COTS spacecraft telemetry data analysis package that provides the user with a set of browsing and data analysis functions for retrieval and decoding of actual telemetry data into statistical subsets of user-defined telemetry data. The ABE package is hosted on the server and is usually associated with EPOCH 2000, allowing access to spacecraft databases for the retrieval of telemetry data, event, and trends data files. The ABE package uses PV-WAVE; a commercial, general-purpose, visual analysis package from Visual Numeric, Inc., PV-WAVE is widely used in the technical community as a foundation for data analysis.

The Jsat trend display is shown in Figure 4. Clicking the Trend menu will launch a graphical user interface (GUI) applet with selectable directories and data subset files. The selected data statistical subset may be viewed with a “View Source” button prior to downloading. The user may select from the following data subset display options: text, plot, scatter plot, bar, area, etc. The statistical data is sent to the text display functions or to a chart in a chart library commercially supplied by the KLG Group for continuous update until the end of the data file.

The ABE remote interface consists of a simple ABE command interface routine written in C. This remote interface processes a profile file that can be generated with any text editor. The file will tell ABE to write the data to a text file to be processed. If the user has limited access only to an offline string, the applets later can be replaced by applications to enable the user not only to view but also to store and manipulate data locally on a user computer.

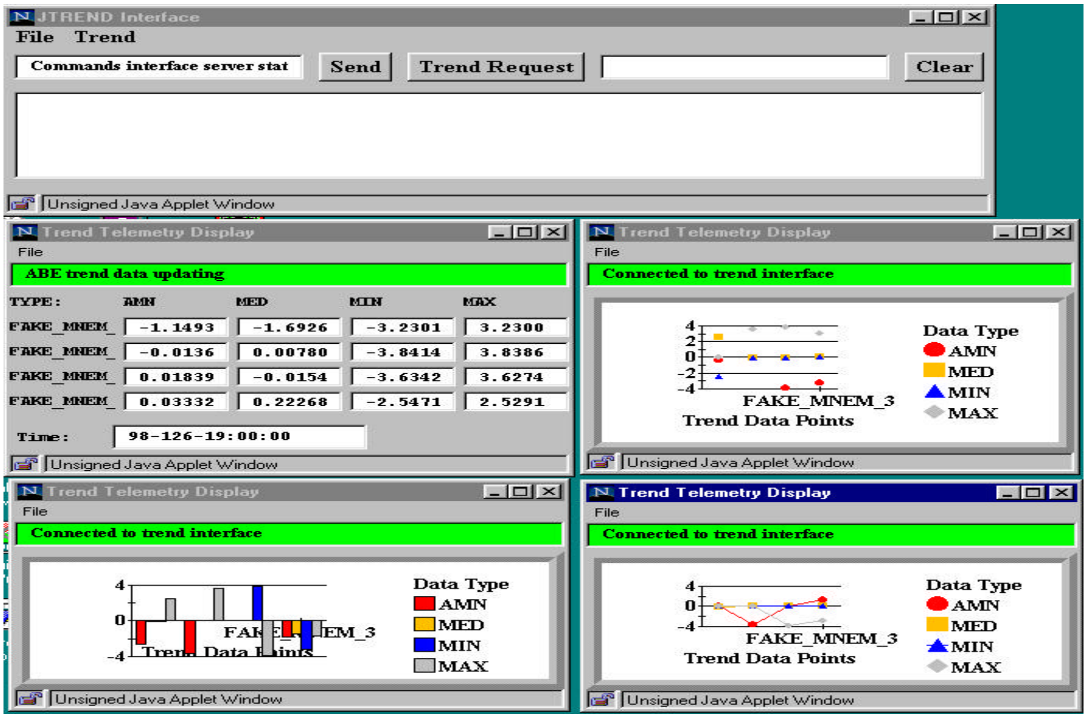
Platform Independence

The Jswitch/Jsat client has been tested on PC, Macintosh, HP, and Sun platforms. Netscape Navigator Gold 3.0, Netscape Navigator Professional Edition 4.04 and 4.6, and Internet Explorer 3.0 browsers have been used. Browser differences were noted in each instance with the rendering of the applets. Differences consisted mainly of font size, widget sizes, and colors. We have standardized by using Netscape Navigator 4.6 for development purposes. On the PC, the primary development platform, these differences were traced to individual preferences in control panel settings.

The colored event applet rendered true red, yellow, or green event colors in Netscape Navigator Gold 3.0 with the Color Palette setting at “256 colors”. However, Netscape Navigator Professional Edition 4.04 rendered the colors incorrectly with the “256 color” setting and correctly with the Color Palette set to “high color 16 bit”.

The trend analysis applet was rendered with the last column in a second row underneath the label column on the developer’s PC and correctly on a second PC with all columns in one row. Turning the screen size on and off corrected this rendering on the developer’s PC.

One exception was noted in the execution of code between the Java Virtual Machine on Internet Explorer 3.0 and Netscape Navigator Professional Edition 4.6. On the Internet Explorer 3.0 Java Virtual Machine, the data values were updated correctly in the



ABE trend data updating

TYPE:	AMN	MED	MIN	MAX
FAKE_MNEM_	-1.1493	-1.6926	-3.2301	3.2300
FAKE_MNEM_	-0.0136	0.00780	-3.8414	3.8386
FAKE_MNEM_	0.01839	-0.0154	-3.6342	3.6274
FAKE_MNEM_	0.03332	0.22268	-2.5471	2.5291

Time: 98-126-19:00:00

Connected to trend interface

Figure 4. Jsat: Trend Applet

labeled text fields; however, the time was incorrectly updated in the Time text field. Netscape 4.6 always displayed the time correctly. The Internet Explorer 3.0 Java virtual machine has incorrectly interpreted the following statement: if (datapoint.time != ("XTIME"). The final solution to this problem probably lies with the Java virtual machine as an add-in to the browser. This product is under development by Sun.

Summary: Jswitch/Jsat fits very well with inexpensive small satellite rapid development

Jswitch/Jsat can quickly provide reusable ground software, with the user evaluating the software from his/her location and providing immediate feedback to the developers. Scientists have a quick, simple access to process their science data. Building ground systems faster is achieved first by using existing software and second by integrating the system using an iterative development process. In the Jswitch prototype systems, several existing COTS products were used. Core command and control functions are readily available in several COTS products (e.g., ISI EPOCH 2000, STI OS/Comet, Altair MCS, LM SCS 21) and Government off-the-shelf (GOTS) products (TPOCC, ASIST). The GenSAA toolkit provides highly graphical expert systems that perform real-time spacecraft monitoring and display. To make use of the WWW, COTS Web servers and browsers were examined that include security features and support the Java virtual machine. The C2net (Apache) Stronghold COTS Web server provides security through encryption and authentication through passwords.

The Jswitch concept enables a better ground system by providing the definition of standard GUI elements that are based on the Java language and applets served up through the Java virtual machine under standard Web browsers. This is a powerful combination because virtually every work location has recent, free Web browsers from Microsoft or Netscape already installed. These products are available for Windows 95, Macintosh, and most versions of UNIX operating systems. In summary, the user benefits from the following capabilities:

- Scientists can communicate with an instrument using a Web browser.
- Engineers and operators can work from office, from home, or on the road using only a Web browser.
- Client computers are platform independent (Web browsers run on PC, Macintosh, and UNIX platforms).
- Anomaly resolution is facilitated, with remote access making lights-out operations during off-shifts realistic.
- Costs are lower; no special hardware or special client software is required beyond a current Web browser.
- Generic bridge interfaces to legacy control centers and telemetry displays are backward compatible.

The use of standard COTS products for major functions provides not only a quicker development cycle but also a less expensive option. This is sometimes mistaken for a panacea. Some COTS products are expensive, and some do not perform as advertised. Obtaining a free trial license for evaluation mitigates this risk. Despite the problems, however, COTS products that can be used in the spacecraft support domain have matured over recent years and present a very good value compared to custom development or even to modification of existing software. The reasons for this include the following:

- Development costs are amortized over many customers.
- The product is enhanced to keep pace with evolution in supporting technologies to remain competitive.
- New features are added as part of product development, also to remain competitive.
- Maintenance costs are usually quoted as a known percentage of initial license costs.
- Special vendor support is usually available for short periods as needed rather than having to maintain dedicated software maintenance staff.

The Jswitch/Jsat prototype system has demonstrated that we can receive telemetry, send commands, process events, display existing legacy telemetry pages, and analyze statistical data over the Web. The Jswitch system can be tailored to interface the Java language through native code to any control center. The Jsat component can easily download science data that has been written to a file. The Flight Operations Team (FOT) and the scientific community can easily be cross-trained on a uniform interface. Mission-specific applications are easily written and downloaded as applets to the user with a standard Java-enabled Web browser. The user always has the latest software updates, and there is no special hardware to buy.